chinaXiv:202310.01507v1

中国搜索 Kubernetes 应用平台部署方案

摘　要：简单地说，kubernetes 是一个管理跨主机容器化应用的系统，是一个容器的编排工具。它实现了包括应用部署、高可用管理和弹性伸缩在内的一系列基础功能，并封装成为一整套完整、简单易用的 RESTful API 对外提供服务。Kubernetes 应用平台的目的是提升服务器性能利用率、高效部署与弹性计算、高冗余、定制化的 PaaS 服务平台。我们希望能创建一套基于 kubernetes 的标准化、模块化和通用性的系统平台，以支撑中国搜索的应用服务。

关键词：Kubernetes；容器化；PaaS 平台；应用服务
中图分类号：TP393　　**文献标识码：**A
文章编号：1671-0134（2019）05-113-05　　**DOI：**10.19483/j.cnki.11-4653/n.2019.05.037

文 / 邵珂　蔡国华　万国雷

1. 项目背景

中国搜索 kubernetes 应用平台项目是一个 PaaS 服务平台，以 kubernetes 为应用基础，整合项目管理系统、CMDB 管理系统以及 CI/CD 的整体运营平台，目的是为向用户提供一个 Docker 应用的统一管理平台，统一分配、管理资源。将 kubernetes 的资源管理接口封装为应用功能，实现可视化 web 管理，极大地降低了 kubernetes 的使用门槛。我们还计划逐步完善平台内的应用生态环境，提供丰富的组件及接口供用户调用。平台自 2018 年 7 月开始研发，10 月投入上线，并持续迭代开发新版本。目前，应用平台支持并提供跨网段计算节点、flannel、calico 网络、万兆网络分布式存储等资源，已有数十个项目在平台上稳定运行。

2.kubernetes 介绍

Kubernetes 是 Google 基于 Borg 开源的容器编排调度引擎，作为 CNCF（Cloud Native Computing Foundation）最重要的组件之一，它的目标不仅仅是一个编排系统，而且提供一个规范，可以让你描述集群的架构，定义服务的最终状态，Kubernetes 可以帮你将系统自动地达到和维持在这个状态。Kubernetes 作为云原生应用的基石，相当于一个云操作系统。Kubernetes 设计理念和功能其实就是一个类似 Linux 的分层架构，如下图所示。

Cloud Native Ecosystem
Interface Layer: Client Libraries and Utilities
Governance Layer: Automation and Policy Enforcement
Application Layer: Deployment and Routing
Nucleus: API and Execution
Container Runtime Interface | Container Network Interface | Container Storage Interface | Image Registry | Cloud Provider | Identity Provider

图 1　kubernetes 分层设计理念

核心层：Kubernetes 最核心的功能是，对外提供 API 构建高层的应用，对内提供插件式应用执行环境。

应用层：部署（无状态应用、有状态应用、批处理任务、集群应用等）和路由（服务发现、DNS 解析等）、Service Mesh（部分位于应用层）。

管理层：系统度量（如基础设施、容器和网络的度量），自动化（如自动扩展、动态 Provision 等）以及策略管理（RBAC、Quota、PSP、NetworkPolicy 等）、Service Mesh（部分位于管理层）。

接口层：kubectl 命令行工具、客户端 SDK 以及集群联邦。

生态系统：在接口层之上的庞大容器集群管理调度的生态系统，可以划分为两个范畴。

Kubernetes 外部：日志、监控、配置管理、CI/CD。

Kubernetes 内部：CRI（容器运行接口）、CNI（容器网络接口）、CSI、（容器服务接口）、镜像仓库、集群自身的配置和管理等。

3. Kubernetes 架构。

3.1 部署方案

高可用集群所需节点配置如下。

角色	数量	描述
etcd 节点	3	为 kubernetes 提供配置管理数据库，要求较高稳定性
master 节点	2	Kubernetes 控制节点，可与 etcd 节点重用，HA+ 主备模式，要求较高稳定性
node 节点	20–100	运行应用负载的节点，可根据需要提升机器配置或增加节点数。要求 CPU 24 核以上，物理内存 64G 以上
存储节点	10–20	网络分布式存储服务的节点，部署 GlusterFS 集群服务，可根据需要扩容节点。要求较高稳定性，万兆网络、大硬盘
Harbor 服务	2	私有镜像仓库服务，HA+ 主备模式，要求较高稳定性

chinaXiv:202310.01507v1

3.2 整体架构

所有服务器被分为三大部分：一部分为 master 服务；另一部分为 node 服务；还有一部分为存储服务。其中，master 服务器因需要实现 LB，需要在同一网段，存储服务器因考虑到效率和稳定性，也需要在同一网段且最好是万兆网络。节点服务器对网络依赖不高，各服务器间路由可达即可。集群全部服务器操作系统均可采用 centos7.4 版本，镜像 yum 源要求 centos 和 epel，各节点时区设置一致、时间同步。

Master 架构包含 etcd 集群和 kubernetes master 服务组件。kuberntes 系统使用 etcd 存储所有数据，是最重要的组件之一，为保证安全性，etcd 需要安装服务器证书，也需要安装客户端证书。

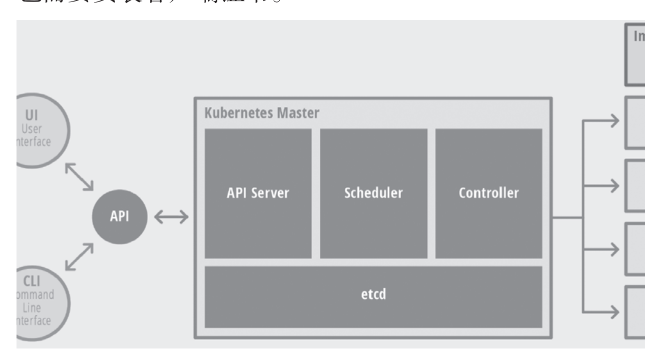


图 2　kubernetes master 服务架构

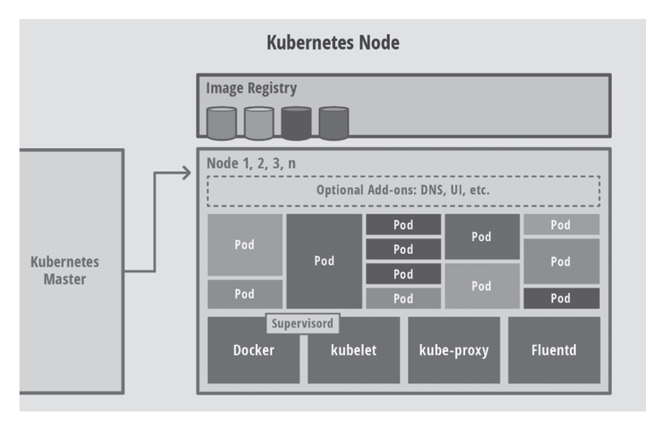


图 3　kubernetes node 服务架构

master 节点主要包含三个组件：apiserver、scheduler controller–manager。其中：

apiserver 提供集群管理的 REST API 接口，包括认证授权、数据校验以及集群状态变更等。

只有 API Server 才直接操作 etcd，其他模块通过 API Server 查询或修改数据，提供其他模块之间的数据交互和通信的枢纽。

scheduler 负责分配调度 Pod 到集群内的 node 节点，监听 kube–apiserver，查询还未分配 Node 的 Pod，根据调度策略为这些 Pod 分配节点。

controller–manager 由一系列的控制器组成，它通过 apiserver 监控整个集群的状态，并确保集群处于预期的工作状态。

master 节点的高可用主要就是实现 apiserver 组件的高可用，通过部署配置 haproxy 对 apiserver 进行负载均衡。

节点服务器主要包含 docker 服务、kuber–node 组件。

3.2.1 Docker

docker 从 1.13 版本开始，将 iptables 的 filter 表的 FORWARD 链的默认策略设置为 DROP，从而导致 ping 其他 Node 上的 Pod IP 失败，因此，必须在 filter 表的 FORWARD 链增加一条默认允许规则 iptables –I FORWARD –s 0.0.0.0/0 –j ACCEPT。

docker 镜像仓库使用国搜内部私有仓库 https://reg.docker.chinaso365.com，可配置在 /etc/docker/daemon.json，替换 docker 的默认仓库。

3.2.2 Kube–node

kube–node 是集群中承载应用的节点，前置条件需要先部署好 kube–master 节点（因为需要操作用户角色绑定、批准 kubelet TLS 证书请求等），它需要部署如下组件。

docker：运行容器。

calico：配置容器网络。

kubelet：kube–node 上最主要的组件。

kube–proxy：发布应用服务与负载均衡。

3.3 网络架构

Kubernetes 基于 CNI driver 调用各种网络插件来配置 kubernetes 的网络，常用的 CNI 插件有 flannel、calico、weave 等，这些插件各有优势，也在互相借鉴学习优点。比如，在所有 node 节点都在一个二层网络时候，flannel 提供 hostgw 实现，避免 vxlan 实现的 udp 封装开销；calico 针对 L3 Fabric 推出了 IPinIP 的选项，利用了 GRE 隧道封装。结合我们的业务需求，采用 calico 网络组建方案。

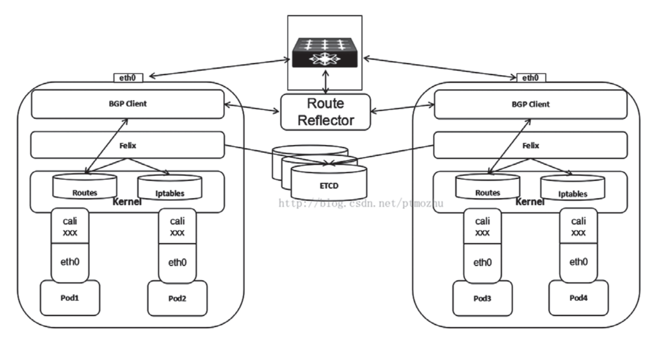


图 4　calico 组网示意

Calico 是一个基于 BGP 的纯三层的数据中心网络方案（不需要 Overlay），并且与 OpenStack、Kubernetes、

chinaXiv:202310.01507v1

AWS、GCE 等 IaaS 和容器平台都有良好的集成。

Calico 在每一个计算节点利用 Linux Kernel 实现了一个高效的 vRouter 负责数据转发，而每个 vRouter 通过 BGP 协议负责把自己上运行的 workload 的路由信息向整个 Calico 网络内传播——小规模部署可以直接互联，大规模下可通过指定的 BGP route reflector 来完成。这样保证最终所有的 workload 之间的数据流量都是通过 IP 路由的方式完成互联。Calico 节点组网可以直接利用数据中心的网络结构（无论是 L2 或者 L3），不需要额外的 NAT、隧道或者 Overlay Network。

此外，Calico 基于 iptables 还提供了丰富而灵活的网络 Policy，保证通过各个节点上的 ACLs 提供 Workload 的多租户隔离、安全组以及其他可达性限制等功能。

3.4 存储架构

GlusterFS 是 kubernetes 私有化部署方案中的最优存储解决方案，它是一个开源的分布式文件系统，具有强大的横向扩展能力，通过扩展能够支持数 PB 存储容量和处理数千客户端。GlusterFS 借助 TCP/IP 或 InfiniBand RDMA 网络将物理分布的存储资源聚集在一起，使用单一全局命名空间来管理数据。GlusterFS 基于可堆叠的用户空间设计，可为各种不同的数据负载提供优异的性能。

我们建议采用分散卷的方式，通过指定冗余块的数量（3:1）达到高可用，同时可提供性价比最优的 IO 吞吐量和网络性能。

性能测试，在万兆网络环境下，IO 约为本地磁盘 IO 性能的 30%，千兆网络环境下对比，接近阿里云 NAS 服务性能的 2 倍。

测试数据如下：

万兆环境	655360 bytes（655 kB）copied, 0.00240202 s, 273 MB/s
千兆环境	655360 bytes（655 kB）copied, 0.0103234 s, 63.5 MB/s
本地磁盘	655360 bytes（655 kB）copied, 0.000753129 s, 870 MB/s
阿里云 NFS	655360 bytes（655 kB）copied, 0.0177044 s, 37.0 MB/s

3.5 集群管理

3.5.1 集群部署

部署方式：ansible 统一部署管理工具，可实现 etcd 服务、docker 服务、master 节点、node 节点、集群网络、证书全自动安装。

集群管理：ansible 统一部署管理工具，可实现 node 节点增、删，master 节点增加 / 替换，etcd 节点增加 / 替换，集群升级，备份恢复。

3.5.2 主要软件及版本

Docker	v18.06.1-ce
Kubernetes	v1.13
Etcd	v3.2.24
Calico	v3.4.1
Coredns	v1.2.6
Glusterfs	v4.1.5

3.5.3 Iptables/IPVS

因为 calico 网络、kube-proxy 等大量使用 iptables 规则，安装前清空所有 iptables 策略规则；CentOS 的 firewalld 等基于 iptables 的防火墙直接卸载，避免不必要的冲突。

kube-proxy 组件监听 API server 中 service 和 endpoint 的变化情况，从而为 k8s 集群内部的 service 提供动态负载均衡。kubernetes 在 v1.10 之前主要通过 iptables 实现，是稳定、推荐的方式，但在当服务多的时候会产生太多的 iptables 规则，大规模情况下有明显的性能问题；在 v1.11 GA 的 ipvs 高性能负载模式，采用增量式更新，并可以保证 service 更新期间连接的保持。

3.5.4 Add-ones

DNS 是 kubernetes 集群首先需要部署的，集群中的其他 pods 使用它提供域名解析服务；主要可以解析 集群服务名 SVC 和 Pod hostname；目前， k8s v1.9+ 版本可以有两个选择：kube-dns 和 coredns，可以选择其中一个部署安装。

Ingress 就是从外部访问 k8s 集群的入口，将用户的 URL 请求转发到不同的 service 上。ingress 相当于 nginx 反向代理服务器，它包括的规则定义就是 URL 的路由信息；它的实现需要部署 Ingress controller(比如 traefik ingress-nginx 等)，Ingress controller 通过 apiserver 监听 ingress 和 service 的变化，并根据规则配置负载均衡并提供访问入口，起到服务发现的作用。

Heapster 监控整个集群资源的过程：首先，kubelet 内置的 cAdvisor 收集本 node 节点的容器资源占用情况；然后，heapster 从 kubelet 提供的 api 采集节点和容器的资源占用；最后，heapster 持久化数据存储到 influxdb 中。

EFG 插件是 kubernetes 项目的一个日志解决方案，它包括三个组件：Elasticsearch、Fluentd、 Grafana。Elasticsearch 是日志存储和日志搜索引擎；Fluentd 负责把 kubernetes 集群的日志发送给 Elasticsearch；Grafana 则是可视化界面查看和检索存储在 Elasticsearch 的数据。

3.5.6 集群升级

集群升级存在一定风险，升级前对 etcd 数据做备份。

快速升级是指只升级 kubernetes 版本，比较常见，

ChinaXiv合作期刊

chinaXiv:202310.01507v1

如 Bug 修复，重要特性发布时使用。快速升级可平滑实现，不会对业务产生中断。

其他升级是指升级 kubernetes 组件，包括 etcd 版本、docker 版本，需制定详细的升级方案和可能的业务中断方案。

4. 中国搜索 Kubernetes 应用平台

中国搜索 kubernetes 应用平台是用于管理数据中心主机集群上的容器化的应用平台，是提升服务器性能利用率、高效部署与弹性计算、高冗余和定制化的 PaaS 服务平台。该平台的目标是让部署容器化的应用简单和高效。

持续部署：平台实现快速、可视化自动部署功能。

弹性伸缩：构建具有需求预测和容器按需供给能力的弹性伸缩子系统。

组件管理：将一个应用涉及的所有组件均做了统一管理。

高可靠性：自动的故障迁移，达到秒级启动，恢复业务。

生态衍生：将可标准化的应用转变为模块化服务，形成应用生态圈。

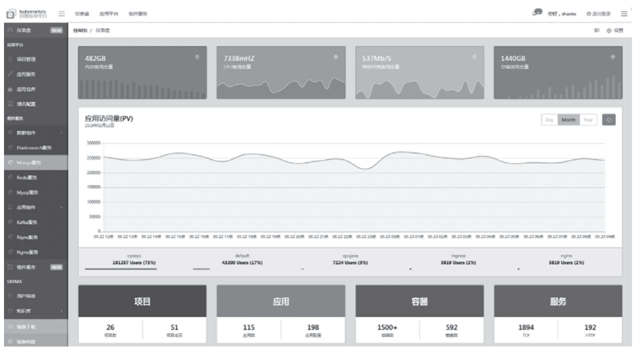


图 5　中国搜索 kubernetes 应用平台－仪表盘（一）



图 6　中国搜索 kubernetes 应用平台－仪表盘（二）

4.1 资源统一分配，实时统计分析

在 kubernetes 集群高效管理的基础上，我们实现了对整体应用情况进行统计分析，相对于传统的物理层监控，应用平台实现了按项目按应用获取、分析、展示资源分配情况、服务健康状态、应用饱和度等应用层面的监控。

通过集群节点状态监控和管理，查看集群服务器运行状态，帮助管理人员实时了解整体情况，及时发现、排查故障隐患。

4.2 项目统一管理

Kubernetes 应用平台采用按项目分配和管理资源及应用的模式，更贴近于常规的运营管理思路，用户按项目为单位在平台上申请权限，获取服务所需资源，包括 CPU 使用量配额、内存使用量配额、存储使用量配额。项目与项目之间相互隔离，并可添加私有镜像仓库认证、kube-api 接口调用认证等访问控制。

项目管理模式实际对应了 kubernetes 中的 namespace，命名空间是一种在多个用户之间划分群集资源的方法（通过资源配额），旨在用于多个用户分布在多个团队或项目中的环境中。

命名空间提供名称范围。资源名称在名称空间中必须是唯一的，而不是跨名称空间。在 Kubernetes 中，在默认情况下，同一名称空间中的对象将具有相同的访问控制策略。

平台支持项目成员管理，在项目管理中可定义多种角色，在命名空间的基础上补充扩展了 kubernetes 对应用管理权限的管理机制。

项目管理员，拥有对项目资源的申请和修改权限。

项目负责人，拥有对项目应用的信息修改权限。

项目成员，拥有对项目应用的访问和控制权限。

4.3 应用服务管理

应用服务管理模块，统一管理项目下的所有应用，是应用配置和访问的入口。包括创建应用服务、查看应用状态，获取应用服务信息。

平台应用服务，即对应调度 kubernetes 中的 POD 服务，是一组一个或多个容器（Docker 容器）化实例，具有共享存储 / 网络，以及如何运行容器的规范。pod 的内容始终位于同一位置并共同调度，并在共享上下文中运行。pod 模拟特定于应用程序的“逻辑主机”，它包含一个或多个相对紧密耦合的应用程序容器。

4.4 支持配置管理、任务管理、监控、日志分析

通过平台提供的应用管理功能，我们可以对 kubernetes 上运行的应用进行标准化的快速配置和应用。通过应用编排实现 kubernetes 应用逻辑的自动配置，通过配置管理实现 configmap 定义或更新，通过任务管理来启动、升级、回滚或停止应用服务，还可通过应用监控实时调用查看应用监控和运行日志。

4.5 支持全应用日志收集

chinaXiv:202310.01507v1

除了 kubernetes 默认收集的 docker 运行日志外，平台还支持用户自定义应用日志的采集，即应用产生的日志文件。通过平台的应用日志收集模块配置分配日志收集服务至项目下，并自动关联项目下应用的存储服务来上传日志。支持多日志格式解析、分类索引配置、自定义索引格式、自定义日志保存时间等。

4.6 支持全容器终端操作

平台支持通过 kubernetes api 访问每个应用容器，即通过 web 终端进入容器内，进行命令行操作，帮助对应用的直观运行控制。容器终端的访问经 kubernetes api 的访问控制及应用平台的权限控制，同时具备高安全性和便捷性。

4.7 支持有状态服务

有状态服务是相对无状态服务的一种 kubernetes 应用部署模式，平台支持无状态应用和有状态应用模式两种应用。使用有状态应用部署模式，应用中的每个 POD 都拥有独立分配且固定的 SVCIP 地址及存储空间，如果容器重启或漂移，它们所使用的资源将不变，使虚拟化更接近于物理层设备。通过有状态应用的模式可在 kubernetes 中实现组建运行，如 redis、kafka、MongoDB 等集群模式服务。

4.8 支持应用备份、复制

平台提供应用备份和复制功能，快速解决应用的重复部署或多环境部署需求。通过应用复制，可快速创建测试环境、预发布环境、生产环境的同步，或者快速部署相同应用需求的组件服务。

4.9 外部访问域名配置管理

平台通过域名配置实现 kubernetes 外部访问应用，通过调用 kubernetes 接口实现 Ingress 配置，提供应用对外服务的出口。通过自定义域名接入内部服务，支持自定义域名配置、路径配置，支持 http 和 tcp 两种模式。

4.10 第三方接口 SDK

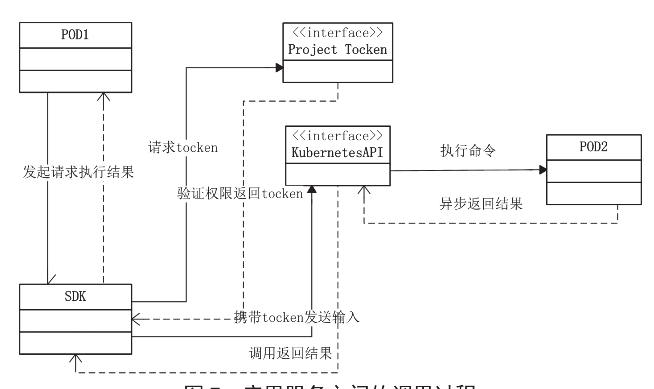


图 7　应用服务之间的调用过程

Kubernetes 平台下运行的应用服务会存在一些在不同的应用间互相调用或状态控制的情况。基于容器间隔离的机制，我们采用 API 实现远程调用，在 kubernetes 应用平台已封装好一套第三方接口服务，满足项目内部的应用间互访需求。平台提供一套统一的客户端 SDK 来满足 Java、Python 工程项目的此类需求。调用流程如下。

容器 a 通过调用 SDK 请求在容器 b 上执行命令，SDK 实现步骤如下：

（1）容器 a 调用 SDK，请求参数包括项目私有 access key 和命令行；

（2）SDK 调用 Project token 接口鉴权，返回鉴权结果，成功返回 token，失败返回错误代码；

（3）SDK 携带 token 请求 kubernetes API，调用接口执行操作命令；

（4）Kubernetes API 在容器 b 上顺序执行操作命令，并异步返回结果给 SDK；

（5）SDK 将执行结果异步返回给容器 a 上的应用。媒

（作者单位：中国搜索信息科技股份有限公司）